UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/904,882	07/12/2001	Calvin White	40101/02401	3600

7590 11/19/2003

FAY KAPLUN & MARCIN, LLP
150 BROADWAY
SUITE 702
NEW YORK, NY 10038

EXAMINER	
VO, TED T	

ART UNIT	PAPER NUMBER
2122	5

DATE MAILED: 11/19/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 09/904,882	Applicant(s) WHITE ET AL.	
	Examiner Ted T. Vo	Art Unit 2122	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *12 July 2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-12* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-12* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *12 July 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. §§ 119 and 120

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

13)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application) since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

a) ☐ The translation of the foreign language provisional application has been received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121 since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____.
4) ☐ Interview Summary (PTO-413) Paper No(s). _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other:

DETAILED ACTION

1. This action is in response to the application filed on 07/12/2001.

Claims 1-12 are pending in the application.

Claim Rejections - 35 USC § 101

2. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefore, subject to the conditions and requirements of this title.

The claims 1-7 are rejected under 35 U.S.C 101 because the claimed invention is directed to non-statutory subject matter.

As per claims 1-3:

The claims 1-3 are rejected under 35 U.S.C 101 because the claimed invention is directed to non-statutory subject matter.

Despite the invention is in technological art, claims 1-3 fail to include hardware to make a tangible system. Moreover, the claim limitations appear comprising software modules that are not tangible in a hardware system.

Analysis: Led by claim 1:

"***A system, comprising:***" (preamble) (fail to include hardware to make a tangible system).

"***a stream source class loader*** *retrieving streaming data to create a desired class object*" (A software element - not be tangible in a hardware system).

"***an interface*** *coupled to the stream source class loader*" (A software element - not be tangible in a hardware system).

"***a plurality of streaming sources*** *containing information including the location of data wherein requests for data are communicated from the stream source class loader to the streaming sources via the interface and, data passes from the stream sources to the stream source class loader via the interface, the streaming sources searching the data locations for the requested data*" (A software element - not be tangible in a hardware system).

Claim 2 and claim 3 are further limitations of the software elements as recited in the claim 1.

To be statutory under 35 U.S.C. 101, each claim, particularly in the preamble, should be amended so that it includes hardware to make a tangible system.

According to analysis: Claims 1-3 are software elements which are not tangible in a hardware system. The claims are merely claiming a program per se which is not statutory under 35 U.S.C. 101.

As per claims 4-7:

The claims 4-7 are rejected under 35 U.S.C 101 because the claimed invention is directed to non-statutory subject matter.

Claims 4-7 are claiming a software program that fails to include functionality so that it could be embedded for causing an execution by a processor.

Analysis: Led by claim 4:

"***A class loader,*** *comprising*" (preamble) (programming procedure).

"***a receiving module*** *to receive a request for a desired class object*": a programming procedure.

"***a stream source module*** *containing information including the location of data which may be streamed to the class loader*" (programming procedure).

"***an interface module*** *which receives requests for data from the receiving module and retrieves the streaming data from the stream source module, wherein the stream source module searches the data locations for the requested data*" (programming procedure).

"***an instantiating module*** *to receive the streaming data from the interface module and instantiate the class object*" (programming procedure).

Claims 5-7 are further limitations of the programming procedures as recited in the claim 4.

To be statutory under 35 U.S.C. 101, each claim should be amended so that its limitation is functional by a processor.

According to analysis: Claims 4-7 are software elements are programming procedures. The claims are merely claiming a program per se which is not statutory under 35 U.S.C. 101.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 1-12 are rejected under 35 U.S.C. 102(e) as being anticipated by Abbott, (US No. 6,542,887).

Given the broadest reasonable interpretation of followed claims in light of the specification.

As per claim 1:

Abbott discloses a JVM that has a class loader (**'Stream Source class loader'**) (figure 1, 'Class Loader'). The class loader responds a request to search and load a native code library (**'streaming data'**) (see column 2, lines 32-35). Generating an object by the class loader from loading the native code library is instantiation (see column 1, lines 10-22, 'instantiate a main class'; see column 2, lines 30-32, 'request to instantiate a class'). The class loader that incorporates (**'Interface'**) LIBPATH and CLASSPATH (**'Streaming Sources'**) searches and loads a requested native code library (see column 2, lines 58-66).

Native code libraries are located in a hierarchy of class files and archived files corresponding to class path sub-directory (see figure 1, and column 2, lines 63-66).

The teaching covers the claim limitations:

"***A system, comprising:***

a stream source class loader (see the figure (fig.1), 'Class Loader' and see column 3, lines 9-13, 'conventional native code library load mechanism') ***retrieving streaming data*** ('native code library') ***to create a desired class object*** (column 2, line 30 'responsive to a request to **instantiate** a class');

an interface coupled to the stream source class loader (See in the figure 1: the path between 'class loader' and 'Class path', 'Libpath'), ***and***

a plurality of streaming sources ('Libpath' and 'Classpath') ***containing information including the location of data*** (see in the figure 1: referring to the arrows pointing to the class file directory and cache, and see column 2, lines 62-66), ***wherein requests for data*** (column 2, line 30 'responsive to a request') ***are communicated from the stream source class loader*** (Class Loader) ***to the streaming sources*** ('Libpath' and 'Classpath') ***via the interface*** (see in figure 1, the arrow started from Class Loader' to 'Classpath') ***and, data passes from the stream sources to the stream source class loader via the interface, the streaming sources searching the data locations for the requested data*** (see figure 1, referring to the arrows from 'Classpath' and 'Libpath' to 'h.jar' and 'Cache'; see column 2 30-42 'search for a second path in which library files are stored').

As per claim 2: Abbott discloses, "***The system according to claim 1, wherein the class object is a Java class object***" (column 3, line 8, 'Java code').

As per claim 3: Abbott discloses," ***The system according to claim 1, wherein the plurality of streaming sources includes one of a zip file stream source, a database stream source and a uniform resource locator stream source***" (see figure 1, '...zip', '...dll', '...jar')

As per claim 4: Abbott discloses, "***A class loader, comprising:***

a receiving module to receive a request for a desired class object (see in the figure 1, 'Class loader');

a stream source module containing information including the location of data which may be streamed to the class loader (see in the figure 1, Classpath and Libpath, and see column 2, lines 58-67, 'A class path is defined to point a list of locations where class files are found');

an interface module which receives requests for data from the receiving module (See in the figure 1, referring to the path between Class Loader and 'Classpath', 'Libpath') ***and retrieves the streaming data from the stream source module*** (See in figure 1, referring to the path connected from 'Classpath', 'Libpath' to class files directories); ***wherein the stream source module searches the data locations for the requested data*** (see column 2, lines 58-67, 'A class path is defined to point a list of locations where class files are found'); ***and***

an instantiating module to receive the streaming data from the interface module and instantiate the class object (column 2, line 30 'responsive to a request to **instantiate** a class').

As per claim 5: Abbott discloses, "***The class loader according to claim 4, wherein the stream source module includes a streaming source set by an application program***" (see column 3, lines 22-28, 'This means that an application can make call').

As per claim 6: Abbott discloses, "***The class loader according to claim 4, wherein the stream source module includes a plurality of streaming sources*** ('Classpath' and 'Libpath') ***wherein the stream source module searches each of the streaming sources to locate the desired data***" (see column 2 62-65, referring to sub-directory').

As per claim 7: Abbott discloses, "***The class loader according to claim 6, wherein the stream source module searches each of the streaming sources beginning with a most likely streaming source and proceeding sequentially in descending order*** (inherently in directory structure) ***of likelihood through the remaining streaming sources***" (see column 2, lines 62-66, 'sub-directory').

As per claim 8: Claim 8 is a method claim that has functionality corresponding to the functionality of claim 1. Claim 8 is rejected in the same reason set forth in connecting to the rejection of claim 1.

As per claim 9: Abbott further discloses the method, "***The method according to claim 8, further comprising the step of: instantiating the class object using the streamed data***" (See column 2, lines 30-31, 'instantiate a class'; and see column 3, lines 4-8, 'load property files and other files used by Java code' (Examiner note: Java code has means of streamed data)).

As per claim 10: Abbott further discloses the method, "The method according to claim 8, further comprising the step of: configuring the stream source including the location of the data for the class Object" (see column 3, lines 9-13, "the conventional native code mechanism...is augmented to search the classpath for the required library, and see column 2, lines 58-67, 'A class path is defined to point a list of locations where class files are found').

As per claim 11: Claim 11 is a method claim that has functionality corresponding to the functionality of claim 3. Claim 11 is rejected in the same reason set forth in connecting to the rejection of claim 3.

As per claim 12: Abbott's class loader is class of Java object-oriented programming. Common knowledge shows a class that consists at least a method. When being executed by a computer, it always returns a result. In the search for a native code library, Abbott assumes a search path is correct (column 3, lines 29-30, 'file names need to be correctly set', line 35, 'the correct file names') and the requested native code library must exist (column 3, lines 45-46, 'If such a DLL is found'). This suggests that an error would be returned from the class loader if the requested native code library were not in the directory. This suggests the teaching of claim limitation: "***The method of claim 8, wherein, if the data associated with the class object is not found, an error is returned***".

Conclusion

4. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Chan et al., US No. 6,470, 944, discloses a dynamicloading of classes in executing a Java program.

Swaminathan et al., EP 0 967 547 A2, discloses a process for creating and attaching a header to Java bytecode.

Liang et al., "Dynamic Class Loading in the Java™ Virtual Machine", 1998 ACM , discloses a method for dynamically loading Java byte code in a JVM. .

Li Gong, "Secure Java Class Loading", IEEE, Pages 56-61, 1998 IEEE, discloses a method to load secure Jave using class loader.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ted T. Vo whose telephone number is (703) 308-9049. The examiner can normally be reached on Monday-Friday from 8:00 AM to 5:30 PM ET. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Dam, can be reached on (703) 305-4552.

The fax phone numbers:

(703) 872-9306 (for formal communication intended for entry);

(703) 746-5429 (for informal or draft communication, please label "PROPOSED" or "DRAFT").

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the Group receptionist whose telephone number is (703) 305-3900.

TED T. VO

Patent Examiner
Art Unit: 2122
November 14, 2003